Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) A processor, comprising:
   decode logic configured to decode instructions from a first instruction set in a first mode and configured to decode instructions from a second instruction set in a second mode, wherein the decode logic is configured to switch from one mode to another temporarily or for a plurality of instructions;
   pre-decode logic coupled to the decode logic and configured to operate in parallel with the decode logic; and
   a status register coupled to the decode logic and configured to indicate whether the decode logic is decoding instructions in the first mode or the second mode;
   wherein the first and second instruction sets each comprise an instruction that temporarily switches the decode logic from one mode to another for at least one subsequent instruction.

2. (Previously Presented) The processor of claim 1, wherein the first and second instruction sets each comprise a Java Impdep1 Bytecode that temporarily switches the decode logic from one mode to another.

3. (Original) The processor of claim 1, wherein the first instruction set comprises a first reserved Java Bytecode that temporarily switches the decode logic from the first mode to the second mode for at least one subsequent instruction, wherein the at least one subsequent instruction belongs to the second instruction set.

4. (Original) The processor of claim 1, wherein the second instruction set comprises a second reserved Java Bytecode that temporarily switches the decode logic from the

second mode to the first mode for at least one subsequent instruction, wherein the at least one subsequent instruction belongs to the first instruction set.

5. (Previously Presented) The processor of claim 1, wherein the first instruction set comprises an instruction that switches the decode logic from the first mode to the second mode for a plurality of instructions.

6. (Previously Presented) The processor of claim 1, wherein the second instruction set comprises an instruction that switches the decode logic from the first mode to the second mode for a plurality of instructions.

7. (Previously Presented) The processor of claim 6, wherein the instruction that switches the decode logic from the first mode to the second mode for a plurality of instructions succeeds a Java Impdep1 Bytecode.

8. (Previously Presented) The processor of claim 1, wherein the second instruction set comprises an instruction that switches the decode logic from the second mode to the first mode for a plurality of instructions.

9. (Currently Amended) A method of decoding instructions from a first instruction set and a second instruction set, comprising:

decoding instructions from the first instruction set in a first mode and decoding instructions from the second instruction set in a second mode, wherein the decoding of both instruction sets is performed on a single decoder;

switching the decoding from one mode to another for one instruction;

switching the decoding from one mode to another for a plurality of instructions[[.]]; and

setting a status register ~~is configured to indicate~~ that indicates whether the decode logic is decoding instructions in the first mode or the second mode.

10. (Original) The method of claim 9, wherein the step of switching the decoding from one mode to another for the one instruction comprises detecting a temporary instruction that indicates the one instruction belongs to another instruction set.

11. (Previously Presented) The method of claim 10, wherein the switching the decoding from one mode to another comprises switching from the first mode to the second mode for the one instruction, wherein the one instruction belongs to the second instruction set.

12. (Previously Presented) The method of claim 10, wherein the switching the decoding from one mode to another comprises switching from the second mode to the first mode for the one instruction, wherein the one instruction belongs to the first instruction set.

13. (Previously Presented) The method of claim 10, wherein the first and second instruction sets each comprise the temporary instruction.

14. (Previously Presented) The method of claim 9, wherein the switching the decoding from one mode to another for a plurality of instructions comprises detecting a temporary instruction that indicates the one instruction belongs to another instruction set.

15. (Previously Presented) The method of claim 9, wherein the switching the decoding from one mode to another for a plurality of instructions further comprises detecting a first permanent instruction that indicates the plurality of instructions belongs to another instruction set.

16. (Original) The method of claim 15, wherein the second instruction set comprises the first permanent instruction.

17. (Previously Presented) The method of claim 15, wherein the switching the decoding from one mode to another for a plurality of instructions comprises switching the decoding from the first mode to the second mode, wherein the plurality of instructions belong to the second instruction set.

18. (Previously Presented) The method of claim 9, wherein the switching the decoding from one mode to another for a plurality of instructions comprises detecting a second permanent instruction that indicates the plurality of instructions belongs to another instruction set.

19. (Original) The method of claim 18, wherein the second instruction set comprises the second permanent instruction.

20. (Previously Presented) The method of claim 18, wherein the switching the decoding from one mode to another for a plurality of instructions comprises switching the second mode to the first mode, wherein the plurality of instructions belong to the first instruction set.

21. (Previously Presented) A processor, comprising:

decode logic configured to decode instructions from a first instruction set in a first mode and configured to decode instructions from a second instruction set in a second mode;

pre-decode logic coupled to the decode logic and configured to pre-decode instructions in parallel with the decode logic wherein the pre-decode logic pre-decodes for a temporary instruction that switches the decode logic from one mode to another for a subsequent instruction; and

a status register configured to indicate whether the decode logic is decoding instructions in the first mode or the second mode;

wherein the first and second instruction sets each comprise the temporary instruction and wherein the second instruction set further comprises a first permanent instruction and a second permanent instruction.

22. (Original) The processor of claim 21, wherein the temporary instruction indicates that the subsequent instruction belongs to another instruction set and causes the decode logic to switch from one mode to another.

23. (Original) The processor of claim 22, wherein the decode logic temporarily switches from the first mode to the second mode, and wherein the subsequent instruction belongs to the second instruction set.

24. (Original) The processor of claim 22, wherein the decode logic temporarily switches from the second mode to the first mode, and wherein the subsequent instruction belongs to the first instruction set.

25. (Previously Presented) The processor of claim 21, wherein the subsequent instruction is the first permanent instruction that switches the decode logic from the first the mode to the second mode for a plurality of instructions.

26. (Previously Presented) The processor of claim 21, wherein the second permanent instruction switches the decode logic from the second mode to the first mode for a plurality of instructions.

27. (Previously Presented) A system, comprising:
main processor; and
co-processor coupled to the main processor, the co-processor comprising:
decode logic configured to decode instructions from a first instruction set in a first mode and configured to decode instructions from a second instruction set in a second mode;

pre-decode logic coupled to the decode logic and configured to pre-decode instructions in parallel with the decode logic, wherein the pre-decode logic pre-decodes for a temporary instruction; and

a status register configured to indicate whether the decode logic is decoding instructions in the first mode or the second mode;

wherein the first and second instruction sets each comprise the temporary instruction, and wherein the second instruction set further comprises a first permanent instruction and a second permanent instruction.

28. (Previously Presented) The system of claim 27, wherein the temporary instruction indicates a subsequent instruction belongs to another instruction set and causes the decode logic to switch from one mode to another.

29. (Original) The system of claim 28, wherein the subsequent instruction belongs to the second instruction set and the decode logic temporarily switches from the first mode to the second mode.

30. (Original) The system of claim 28, wherein the subsequent instruction belongs to the first instruction set and the decode logic temporarily switches from the second mode to the first mode.

31. (Previously Presented) The system of claim 28, wherein the subsequent instruction is the first permanent instruction, wherein the first permanent instruction indicates a plurality of instructions from the second instruction set is to follow and switches the decode logic from the first mode to the second mode for the plurality of instructions.

32. (Previously Presented) The system of claim 27, wherein the second permanent instruction indicates a plurality of instructions from the first instruction set is to follow and

switches the decode logic from the second mode to the first mode for the plurality of instructions.

33. (Original) The system of claim 27, wherein the system comprises a cellular telephone.

34. (New) The processor of claim 21 wherein the decode logic is configured to decode instructions from the first instruction set being a stack-based instruction set, and is further configured to decode instructions from the second instruction set being a register-based instruction set.